CLAIMS

1. In a Java computing environment, a Java macro instruction representing:

a sequence of Java Bytecode instructions consisting of a Java Getfield Bytecode instruction immediately followed by a Java Astore Bytecode instruction,

wherein said Java macro instruction can be executed by a Java virtual machine operating in said Java computing environment, and

wherein, when said Java macro instruction is executed, the operations that are performed by said conventional sequence of Java Bytecode instructions are performed.

2. A Java macro instruction as recited in claim 1, wherein said Java macro instruction consists of a conventional Java Getfield Bytecode instruction immediately followed by a conventional Java Astore Bytecode instruction.

3. A Java macro instruction as recited in claim 1, wherein said Java macro instruction is generated during the Java Bytecode verification phase.

4. A Java macro instruction as recited in claim 1, wherein said Java virtual machine internally represents Java instructions as a pair of streams.

5. A Java macro instruction as recited in claim 4,

wherein said pair of streams includes a code stream and a data stream,

wherein said code stream is suitable for containing a code portion of said Java macro instruction, and

wherein said data stream is suitable for containing a data portion of said Java macro instruction.

6. A Java macro instruction as recited in claim 5,

wherein said Java macro instruction is generated only when said virtual machine determines that said Java macro instruction should replace said conventional sequence.

7. A Java macro instruction as recited in claim 6, wherein said determination is made based on a predetermined criteria.

8. A Java macro instruction as recited in claim 7, wherein said predetermined criteria is whether said conventional sequence has been repeated more than a predetermined number of times.

9. In a Java computing environment, a Java macro instruction representing:

a sequence of Java Bytecode instructions consisting of an inventive Java Getfield Bytecode instruction immediately followed by an inventive Java Astore Bytecode instruction,

wherein said Java macro instruction can be executed by a Java virtual machine operating in said Java computing environment, and

wherein, when said Java macro instruction is executed, the operations that are performed by said sequence of Java Bytecode instructions are performed.

10. A Java macro instruction as recited in claim 9,

wherein said inventive Astore instruction operates to store values located on an execution stack into arrays, the virtual machine instruction representing two or more Java Bytecode executable instructions that are also suitable for storing values located on an execution stack into an array.

11. A Java macro instruction as recited in claim 10, wherein the arrays can be an array of 1 byte values, 2 byte values, 4 byte values, or 8 byte values.

12. A Java macro instruction as recited in claim 11, wherein a header of an array is read to determine the type of the array.

13. A computer readable media including computer program code for a Java macro instruction, said Java macro instruction representing:

a sequence of Java Bytecode instructions consisting of a Java Getfield Bytecode instruction immediately followed by a Java Astore Bytecode instruction,

wherein said Java macro instruction can be executed by a Java virtual machine operating in said Java computing environment, and

wherein, when said Java macro instruction is executed, the operations that are performed by said conventional sequence of Java Bytecode instructions are performed.

14. A computer readable media as recited in claim 13, wherein said Java macro instruction consists of a conventional Java Getfield Bytecode instruction immediately followed by a conventional Java Astore Bytecode instruction.

15. A computer readable media as recited in claim 14, wherein said Java macro instruction is generated during the Java Bytecode verification phase.

16. A computer readable media as recited in claim 15, wherein said Java virtual machine internally represents Java instructions as a pair of streams.

17. A computer readable media as recited in claim 16,

wherein said pair of streams includes a code stream and a data stream,

wherein said code stream is suitable for containing a code portion of said Java macro instruction, and

wherein said data stream is suitable for containing a data portion of said Java macro instruction.

18. A computer readable media as recited in claim 17,

wherein said Java macro instruction is generated only when said virtual machine determines that said Java macro instruction should replace said conventional sequence.

19. A computer readable media as recited in claim 18, wherein said determination is made based on a predetermined criteria.

20. A computer readable media as recited in claim 19, wherein said predetermined criteria is whether said conventional sequence has been repeated more than a predetermined number of times.